CLAIMS

1. A method of queuing variable size data packets in a communication system, the method comprising:

generating from a said data packet a record portion of predetermined fixed size and containing information about the packet, the data in the packet being in a data portion;

storing data portions in independent memory locations in a first memory with each data portion having no connection with any other;

storing record portions in one or more managed queues in a second memory having fixed size memory locations equal in size to the size of the record portions;

wherein:

the first memory is larger and has a lower address bandwidth than the second memory; and

the memory locations in the first memory are arranged in blocks having a plurality of different sizes and the memory locations are allocated to the data portions according to the size of the data portions.

2. A method as claimed in claim 1, wherein there are two sizes of memory location in the first memory arranged in two said blocks, one of a size to receive relatively small data portions and the other of a size to receive relatively large data portions, and wherein data portions that are too large to be stored in a single memory block are stored as linked lists in a plurality of blocks with pointers pointing to the next block but without any pointers pointing from one data portion to the next data portion of the packet.

3. A method as claimed in claim 1 or claim 2, wherein the sizes of the memory locations in the blocks are matched to the most commonly occurring sizes of data packets in the communication system.

4. A method as claimed in any one of claims 1 to 3, further comprising allocating the memory locations in said first memory from a pool of available addresses provided to it in batches from a central pool of available addresses.

5. A method as claimed in claim 4, wherein the memory blocks are segregated into a plurality of memory channels, the method further comprising allocating addresses to data portions sequentially across channels whereby to spread the storage of data portions across the channels.

6. A method as claimed in claim 4, further comprising reading the data portions from the first memory in pipelined manner by a data retrieval unit adapted to instruct a memory block to read out a data portion without having to wait for the previous read to be completed, and releasing the address location from the first memory.

7. A method as claimed in any of the preceding claims, further comprising, under circumstances where there is insufficient memory for a received packet, enqueueing the record portion as though the corresponding data portion was stored in the first memory, subsequently reading out the record portion corresponding to the said data packet, setting a flag to indicate that the data portion of the said packet is to be discarded, discarding the said data portion, and releasing the memory location notionally allocated to the discarded data portion.

8. A method as claimed in claim 6, further comprising reading the address locations from a bitmap of addresses and, when a memory location is released after the data stored therein has been read out, the address of the released memory location is sent directly to the pool.

9. A memory hub for queueing received data packets, comprising:

an arrivals block, adapted to generate from a said data packet a record portion of predetermined fixed size and containing information about the packet, the data in the packet being in a data portion;

a first memory for storing data portions in independent memory locations, with each data portion having no connection with any other;

a second memory for storing record portions in one or more managed queues, the memory having fixed size memory locations equal in size to the size of the record portions;

wherein:

the first memory is larger and has a lower address bandwidth than the second memory; and

the memory locations in the first memory are arranged in blocks having a plurality of different sizes and the memory locations are allocated to the data portions according to the size of the data portions.

10. A memory hub as claimed in claim 9, wherein there are two sizes of memory location in the first memory arranged in two said blocks, one of a size to receive relatively small data portions and the other of a size to receive relatively large data

portions, and wherein data portions that are too large to be stored in a single memory block are stored as linked lists in a plurality of blocks with pointers pointing to the next block but without any pointers pointing from one data portion to the next data portion of the packet.

11. A memory hub as claimed in claim 9 or claim 10, wherein the sizes of the memory locations in the blocks are matched to the most commonly occurring sizes of data packets in the communication system.

12. A memory hub as claimed in any one of claims 9 to 11, wherein the memory locations in said first memory are allocated from a pool of available addresses provided to it in batches from a central pool of available addresses.

13. A memory hub as claimed in claim 12, wherein the memory blocks are segregated into a plurality of memory channels, and addresses are allocated to data portions sequentially across channels whereby to spread the storage of data portions across the channels.

14. A memory hub as claimed in claim 12, further comprising a data retrieval unit adapted to read out the data portions from the first memory in pipelined manner and to instruct a memory block to read out a data portion without having to wait for the previous read to be completed, and releasing the address location from the first memory.

15. A memory hub as claimed in any of claims 9 to 14, further comprising flag setting means such that, under circumstances where there is insufficient memory for a received packet, the record portion is enqueued as though the corresponding data portion was stored in the first memory, the record portion corresponding to the said data packet is subsequently read out, and the flag setting means sets a flag to cause the data portion of the said packet to be discarded and the memory location notionally allocated to the discarded data portion released.

16. A memory hub as claimed in claim 14, further comprising a bitmap of address locations and means operable such that, when a memory location is released after the data stored therein has been read out, the address of the released memory location is sent directly to the pool.